Test and Training ENabling Architecture (TENA)

TENA BASELINE PROJECT REPORT

Volume II

Product Line Approach

A Cooperative Methodology for Supporting Test and Training Resources and Ranges

Under Secretary of Defense (Acquisition & Technology)

Director, Test, Systems Engineering and Evaluation

Office of Resources and Ranges

Central Test & Evaluation Investment Program

Pentagon (Room 3D1067)

Washington, DC 20301-3110

September, 1997

Abstract

The Test and Training ENabling Architecture (TENA) Product-Line Approach (PLA) presents a Cooperative Methodology for Supporting Test and Training Resources and Ranges. The PLA is fundamental to engendering the cost savings required of DoD ranges in the future, and breaking the paradigms of the present. The TENA Architecture will provide significant cost savings through interoperability, reuse, and sharing, but the true breakthrough or revolution in the effectiveness and economics of test and training ranges and resources can only be realized through a fundamental shift in the way we design, develop, deploy, and maintain these systems. That revolution is called the Product-Line Approach.

Radically different techniques are needed to meet the demand for increased software functionality, at a time when DoD has less money and staff to accomplish the task. The Product-Line Approach (PLA), makes sound technical and financial sense for the Test and Training communities. The Product-Line Approach offers specific advantages over the current project-oriented development strategy. Development time and cost are significantly reduced. Products are engineered through recognition of changes within fundamental requirements or product-line architectures, rather than built from scratch. In addition, under the PLA, the range community can provide specific guidance to suppliers for vendor qualifications, development standards, and product definitions.

By implementing a PLA and serving 10 range sites we will save at least 207 million dollars in software development costs and 543 million dollars over ten years in cumulative development and maintenance cost. These savings compare to experiential data from product-line success stories.

This document describes the Product-Line Approach and defines the organizations required to implement it and the processes used by those organizations. The application to ranges is intuitive, but will require additional research to establish the best organization and processes.

An incremental, evolutionary transition to a Product-Line Approach is recommended as an integral part of TENA’s response to realizing the goals and objectives at the highest levels of DoD for modernizing our test and training infrastructure.

The opinions, ideas and recommendations presented in the TENA Baseline Project Report are the views of the TENA Project Team and do not necessarily represent those of the Sponsor.

Table of Contents

List of Figures and Tables

Purpose

The Product-Line Approach (PLA) presents a cooperative methodology for supporting test and training resources and ranges. The reader is introduced to the PLA organization and processes. A rationale for change leads to a convincing discussion on the need for shifting to a Product-Line Approach for range systems development. Two industry examples in Appendix C show clearly the extensive advantages provided by a shift to a PLA organization. An analysis of the advantages and challenges of the new approach and some implementation issues are also presented.

Readership

The Product-Line Approach is intended primarily for range management and operation directors and others in an range management oversight or DoD and Service decision making role. It also contains information of interest to software professionals involved with test and training ranges and facilities. The PLA represents a new concept designed for software development and maintenance, but is also applicable to hardware development, deployment and support. The intent is to introduce and discuss the concepts required to understand the PLA. It is not intended to be an implementation or transition plan. It does not provide managers with the detailed steps involved in planning for the transition, including establishing accountability, managing risk, scheduling, and budgeting. However, it does offer a clear methodology to realizing the goals and objectives at the highest levels of DoD for modernizing our test and training infrastructure.

Relationship to Other Volumes

Appendices contain information from the Software Engineering Institute [Brownsword, 1996] and elsewhere [Dikel, 1997] [Macala, 1996] which provide in-depth discussion on the Product-Line approach.

In other volumes of the TENA Baseline Project Report we present the Technical Reference Architecture (TRA), TENA Application Concepts and our proposed Logical Range Business Process Model. We introduce the concept of the Logical Range as a means of explaining the object-oriented architecture approach. The TRA (Volume IV) is built to support the Product-Line Approach and the Transition Plan (Volume VII) identifies issues and plans for implementing the TRA.

PROJECT NEED

TENA is part of a coordinated response by the Central Test and Evaluation Investment Program (CTEIP) office to several current and emerging challenges in the test and training range and resource community. These challenges include:

- Reducing software development and maintenance cost,
- Utilizing common instrumentation at multiple facilities,
- Responding to the increased demand for multiple-site exercises and/or exercises which cross T&E/training or live/virtual/constructive boundaries,
- Responding to the increased demand for consistency of information between facilities and across phases of the acquisition process, and
- Capturing critical data to support informed customer and management decisions about resource needs, capabilities, and investments.

PROJECT PURPOSE

The purpose of the TENA project is to respond to these challenges through the establishment of an architecture that efficiently and effectively fosters the sharing, reuse, and interoperability between cooperating Department of Defense (DoD) test ranges and facilities, training ranges, laboratories, and other modeling and simulation activities. The expected synergism will permit efficient and effective testing of new and enhanced weapons systems and will vastly improve the scope and fidelity of worldwide joint/combined training.

PROJECT HISTORY

The Test and Training ENabling Architecture (TENA) project concept was formulated in FY95 by a multi-Service working group. This concept was endorsed by the Test and Evaluation Reliance Investment Board (TERIB), the Board of Operating Directors (BoOD), and the Test and Evaluation Resource Council (TERC).

The Navy is the CTEIP Resource Manager for this project, and has established a Joint Project Office (JPO) for the management of project activities at the Naval Undersea Warfare Center (NUWC) Division, Newport, RI.

Shortly after assembly of the Joint Service Team, several critical observations were made:

- The key to interoperability is not connectivity alone, but rather understanding communications content. This is best promoted by defining an open, object-oriented software architecture that could be used by both legacy and newly built systems.
- The process used to plan, schedule, and otherwise coordinate a multiple-facility, multiple-service exercise must be integral to the development of the architecture, or the capabilities it offers might never be fully utilized.
- The architecture must be conducive to refinement over time and coexists with facility-unique applications. This requires a disciplined architecture development/refinement

process. The team adapted the Defense Information Systems Agency (DISA) domain-engineering approach to help develop the architecture and recommends the Product-Line Approach for implementation and life-cycle maintenance.

- Significant investments are being made in other closely related areas such as, Defense Modeling and Simulation Office (DMSO), High Level Architecture (HLA) and the Joint Simulation system (JSIMS) program. TENA must leverage as many of these efforts as practical.
- The TENA concept is radically new to our community. Planning for transition is key to its ultimate acceptance.

STATUS

The project team tested its architecture development process in FY96 producing a "Pilot Architecture." This work was reviewed in several public forums. These reviews were highly supportive of TENA’s effort. Two consistent suggestions were that TENA should focus first "on breadth, not depth", and that there should be more emphasis on "problem-space vs. solution-space". These considerations and additional engineering effort has resulted in this refined "Baseline Architecture."

The TENA Baseline contains sufficient detail to continue further analysis and risk reduction efforts and is a good vehicle for discussion, experimentation, and refinement. It is not yet appropriate to use these documents as the blueprint for a major system development. After community feedback, results from risk-reduction prototypes, experiments, and other ongoing efforts are synthesized, the cognizant TENA Baseline documents will be updated as "TENA Rev 0." TENA Rev. 0 will be the appropriate source of design information for a TENA-compliant system implementation.

.

A product-line for software is a collection of software systems that addresses a common set of system requirements for a particular business activity or mission. The development of software systems in the product-line is characterized by the use of common assets including product-line architectures, components, and process models. Products in the product-line are built using these common assets, plus some system-unique software.

Using the Product-Line approach, the range development organization (called the range developer) works directly with the range users and sponsors in making the decisions on the requirements and needs of a particular range system. A TENA-compliant system offers access to both organic and distributed resources, giving the developers and sponsors flexibility in deciding which capabilities must be built locally and which they can more cost-effectively achieve by interoperability with other systems. The sponsor remains involved throughout the development and can monitor and validate the development of the range system as it evolves from prototype to final deployment (Figure 1, Part A).

Rather than building from scratch, the range developer engineers products in the product-line through customization from base requirements and a standard product-line architecture based on facility requirements. The development group integrates components and system-unique software using generator and manual techniques. Figure 1, Part B, illustrates this concept.

The standard architecture for range products is called the Test and Training ENabling Architecture [TENA,1]. The basic problem articulated for TENA is really more substantial than just defining the architecture. The effectiveness of the architecture will be multiplied many times over if it is deployed as part of the Product-Line Approach. It is recognized that the manner in which test and evaluation is performed today is simply too expensive [Sanders, 1997]. Although undoubtedly the result of numerous causes, the duplication of resources and creation of multiple systems that provide the same or similar capabilities are major contributors. The redundancy in systems is evidenced by multiple systems at separate test and training facilities performing similar functions. In addition, multiple systems are created sequentially for individual ranges that provide the same or increased functionality as existing systems without reusing an economically significant amount of existing systems resources. In other words, all systems are built from scratch, even though they duplicate in part or in entirety other systems.

Figure 1. Product-Line Approach for Range Development

TENA will foster efficient and effective sharing, reuse, and interoperability among resources within the test and training communities. Adoption of TENA will reduce operating costs through an open systems architecture, employment of advanced digital electronics developments, application of distributed interactive simulation techniques, and use of commercial-off-the-shelf (COTS) technologies where applicable. The use of the architecture will support evolution of a set of test and training assets that reduce or eliminate redundant facilities and assets. This approach to test and training system implementation affords widespread reuse of systems and system components.

Using PLA, the range development organization performs the following tasks:

- Utilizes the core architecture for all range-related products,
- Develops range-unique component assets for that architecture,
- Provides range products to range customers, and
- Supports the implementation and maintenance of the development and execution environments for ranges.

There are four groups, shown in Figure 2, within the range development organization:

- The Architecture Group produces the TENA product-line architecture definition (Technical Reference Architecture and domain specific architectures) for all range development organization products. The architecture group also collaborates in building specific applications by recommending use of product-line assets to the range product development groups based on user requirements and by analyzing needs and tailoring the product-line architecture for production of the application.

- A Component Asset Group develops assets within specific areas of range expertise for use in range products. The asset group also defines and evolves product-line architectures with the Architecture Group.

- The Product-Line Support Group defines the development and execution environments for range products.

- Range Product Development Groups develop and deliver range products for users in the field. They develop a system architecture using the product-line architecture, including the technical architecture and components. These groups will generally be located at major range facilities or Service laboratories.

Figure 2. Proposed Range Product-Line Organizational Structure

This functional grouping should not be interpreted as a recommended organizational structure. Specifically this should not be interpreted as a way to geographically distribute work to multiple locations. The range development organization must function as a cohesive unit with strong group interaction and feedback. Although some functions can be accomplished in a distributed manner, the groups we define are only convenient ways of discussing the prime functions of an integrated team.

RATIONALE FOR CHANGE

With the current acquisition process, it is not unusual for major systems to require 7 to

10 years to progress from conceptualization through research and development, design, integration, and test, to deployment. We are continuing to relearn lessons in each development, and we are not taking advantage of improved reliability, common operations, and training.

Radically different approaches are needed to meet the demand for increased software functionality at a time when the Department of Defense has less money and staff to accomplish this task. New techniques, such as the Product-Line Approach, can be applied to meet these challenges.

The Product-Line Approach can offer specific advantages over a project-oriented development strategy. [Brownsword, 1996] Development time and cost are significantly reduced. Organizations build core competencies, which are concentrated areas of knowledge that allow them to make more productive use of their staff. Products are engineered through recognition of changes within fundamental requirements or product-line architectures, rather than built from scratch. In addition, under the Product-Line Approach, the range community can provide specific guidance to suppliers for vendor qualifications, development standards, and product definitions.

The Product-Line Approach to developing and maintaining DoD systems is supported by the Office of the Secretary of Defense. The Air Force is currently planning to implement product-lines, consistent with direction and guidance from the DoD. A product-line strategy is consistent with and complements the ongoing acquisition reform and streamlining initiatives within the DoD and Air Force. [Perry 1994] [Lightning Bolt] [Dikel, 1997] [Macala, 1996]

By exploiting commonalties and controlling the variability across related systems, the range community can develop strategies that will enable the fielding of systems faster, cheaper, and with added capability for the T&E and training. For the product-line concept to work, there is a fundamental change required in the way system requirements are defined. Range developers must be aware that they will be called upon to decide on the tradeoffs associated with the elimination or modification of some requirements.

Within this constraint, the Product-Line Approach will result in:

- Consolidation of core resources and competencies through identification of key business areas,
- Increased quality through the use of assets that are well understood and proven through retesting during multiuse,
- Building of tailorable features into assets to meet more than one user's needs,
- Minimizing of number of assets--reducing overall and repetitive development costs,
- Reduction of risk in software performance through known performance of assets,
- Improved time to production through reuse of technology, design, and assets,
- Increased interoperability through reuse of common architectures, interfaces, and

protocols, and

- Reduced training requirements for operations and Operation & Maintenance (O&M) through similarities of components.

This section presents the concepts for:

- Role of architecture,
- Management of component assets,
- Development and execution environments, and
- Development of systems in the product-line.

This section also includes scenarios for product-line asset development and product-line system production.

THE ROLE OF ARCHITECTURE

The architecture is critical to the success of the Product-Line Approach. Designers create system parts within the structure provided and the constraints imposed by the architects. The discipline is enforced, and compliance is mandatory for designers and implementers. The architecture remains, throughout the life of the system, an accurate conceptual model of the structure of the system, and it is adapted as required based on information discovered during design and implementation. Key product-line decisions are made during the process of developing or selecting the product-line architecture. These decisions are based on the following questions/issues:

- What are the critical issues in product-line development (product-line selection and inclusion, handling commonalties and differences, security, interoperability, reliability in product delivery)?
- How will the product-line support interoperability/component integration issues (e.g., the High Level Architecture (HLA)?
- What are the plans for compliance and levels of compliance?
- Whether and how to support legacy systems.
- New development support requirements.
- What are the plans for change/evolution management within the product-line?
- What are the key quality factors (e.g., for example, performance, security, dependability) that are essential for the product-line?
- How will the product-line take advantage of Commercial-off-the-Shelf (COTS)/software sharing?

Before institutionalizing the TENA architecture, the test and training community must take an enterprise-wide look at its products. A first step is segmenting these products into product-lines through an identification and scoping process. Mission area analysis to define the organization's business and the development of the organizational structure for product-line development is a part of this step. The next steps in the decision process include product-line specification, and development of a technical reference architecture to guide the development of system architectures for individual products.

- *Specification of the product-line.* Specification requires understanding the potential commonalities across current and future systems in the product-line as well as variations that lead to different systems. This key step requires analysis of product-line capabilities, those that are mandatory for each system in the product-line, and those that may be optional. In addition, the definition must provide for alternative capabilities, i.e., a choice among different capabilities, where appropriate.

- *Development of the TENA Technical Reference Architecture.* The Product-Line Architecture defines the components (mandatory, optional, alternative), component interrelationships, constraints, and guidelines for use and evolution in building systems in the product-line. The product-line architecture must support common capabilities identified in the specification and the potential variability within the product-line. It will be based upon the technical reference architecture and architecture components (services) that will be used in new and updated product-line systems. The reference architecture supports the development of system architectures for domain classes of test and training facilities, e.g., Open Air Ranges (OARs), Hardware-in-the-Loop (HITL) facilities, Installed System Test Facilities (ISTFs), as well as for components used in implementing individual range systems (see Figure 3). Architecture guidelines will discuss factors involved in the use and evolution of the architecture.

- *System architectures for classes of systems.* The Technical Reference Architecture provides common services and guidelines for the basic classes of test and training facilities. While OARs, HITLs, ISTFs, etc., all share these common services, they will differ in specific components. In addition to services, the TRA provides the architecture basis for these components. The system architectures for each class of facilities serves as the basis for implementing individual range systems for specific exercises or missions. The developers apply domain specifics and constraints using the TRA, as shown in Figure 3. The vertical lines show the most active areas for each organization during the development process.

Figure 3. Architecture Process

- *System architectures for individual products.* Representatives from the product-line organizations form a product-line architecture selection team. This team collaborates in product-line production to determine architecture suitability for a new system. The team must assess the ability of the product-line architecture to meet the specific system needs as defined by the user. This architecture assessment considers existing products in the product-line, as well as architectural constraints.

  Existing products may serve as a model for the new system, or the product-line assets may support a prototyping capability. The architecture team must determine if the needs of the new system can be met within the current product-line architecture. If not, they must decide:

  - Whether the system needs can be relaxed, so that the product-line architecture can be used, and

  - Whether it is feasible to use parts of the product-line architecture or to extend it for this new need and for future systems in the product-line.

The selection team may decide that system development cannot be performed with the Product-Line Approach and then employ alternate acquisition methods.

The Architecture Group will work together with range users to develop the TENA architecture based on user needs. The user (customer) will drive the PLA concept, as well as the new Logical Range way of doing business in the future. TENA is designed to enable that shift in methodology.

SYSTEM DEVELOPMENT CONTEXT

Product-line assets are the reusable resources that support the development of products in a product-line. These assets are more than just software components. They include:

- *Domain models*
- *Domain knowledge*
- *Product-line architectures*
- *Test plans and procedures*
- *Communication protocol descriptions*
- *Requirement descriptions*
- *User interface descriptions*
- *CM plans and tools*
- *Code components*
- *Performance models, metrics*
- *Work breakdown structures*
- *Budgets and schedules*
- *Application generators*
- *Prototypes*
- *Process components (methods, tools)*
- *COTS product profiles*
- *Designs, design standards, design decisions*
- *Test scaffolding*

Each development cycle of a system in the product-line offers an opportunity to refine these assets.

The activities required to identify and maintain product-line assets include:

- Identifying, qualifying, and packaging reusable resources (enterprise-wide assets) for use in future development,
- Making them available within and across range product-lines (through a repository and other communication channels), and
- Maintaining configuration control on versions.

Furthermore, in the case where a product-line has made the commitment to leverage commercial investment by focusing on the integration of COTS products as a development method, it will be necessary to have the infrastructure in place to,

- Perform suitability testing of COTS products using a centrally maintained facility.

The Component Asset Group is primarily responsible for performing these tasks under the Product-Line Approach. However, the asset group is supported by the other product-line organizations. For example, in identifying enterprise-wide assets, the architecture group will play a major role as part of its task in developing product-line architectures. This is especially the case for range-specific assets. For COTS products, the asset group will remain the major source for identifying and determining suitability of assets.

Identify Enterprise-Wide Assets

An important core product-line effort involves the identification of reusable assets for use within and across product-lines and the development of a reusable asset base. Legacy systems must be analyzed to identify existing software for possible use as reusable information and assets. Assets from legacy systems and new development include software, architectures, designs, criteria, and other information. This information will be maintained in a product-line asset repository. Identification and packaging of these enterprise-wide assets will increase the asset base available to each product-line organization.

Another ongoing task to support the identification and distribution of enterprise-wide assets is cross-product-line analyses of these assets to identify opportunities for reuse of products and knowledge in other product-lines. Technology transfer of this information, as well as emerging reuse techniques and methods across product-lines, will be performed to maximize the benefits of the opportunities identified.

Repository

A repository of product-line information acquired through suitability testing and identification of enterprise-wide assets activities will be maintained. This will include all of the kinds of assets stated above, organized according to product lines. Range-sensitive information will be available through an access-controlled repository. A list of the products tested and the results of suitability testing will be made available through a separately maintained Product List. Eventually, an acquisition mechanism for COTS products may be provided in addition to the Product List.

The asset repository will accelerate and support availability of proven, reliable assets for incorporation into product-line systems. As the repository is fully populated and the working relationships among the organizations mature, the opportunities for reuse will increase and the benefits of the Product-Line Approach will be realized.

Suitability Testing

Suitability testing is the process of determining if a COTS or Government off-the-shelf (GOTS) software product meets the architectural and functional requirements of a component area within a software architecture. The products are tested using a standard process to provide an objective analysis of the functionality and architectural capabilities using criteria that are derived from the architecture. COTS and GOTS products will be tested for suitability against product-line architectures. Suitability criteria will be developed and maintained. The suitability criteria are derived from requirements and interfaces for component areas in product-line architectures and are used to perform suitability testing of software products. Results of suitability testing will be placed in the Approved Product List.

DEVELOPMENT AND EXECUTION ENVIRONMENT

The Support Group is responsible for producing environments for product-line asset development and execution support of range products. Working with the Architecture Group and the users, the Support Group defines three areas:

- *The development environment* - This includes the software development, test, integration, and maintenance environments, from development through installation. The architecture and component asset groups use this environment to develop product-line assets; the product development groups use them to produce range products for users.

- *The execution environment* - This environment defines hardware/software integration. It establishes actual system behavior in terms of interactions between range products produced from the product-line with other range assets. The execution environment also supports performance analysis based on the use of specific combinations of component assets within the TENA architecture.

- *The support environment* - In some product-lines, individual systems are deployed through the support environment where the user provides parameters to define system operations, user characteristics, and system environments. The support environment delivers an operational system. Variations among systems in the product-line systems may result in differing support environments for development, prototyping, etc.

These same parameters also define the variability within the potential host environment. For systems in a product-line, there may be a variety of potential host configurations. Analysis of the product-line should define the various customer sites and interfacing systems. There may be specific COTS systems interfacing with the system. Variations in systems operations and assumed workloads on a host may be factors in determining architecture for specific operational capabilities.

DEVELOPMENT OF SYSTEMS IN A PRODUCT-LINE

The process for developing systems with a Product-Line Approach differs from the current process in two ways. These are:

- *Development from standard architectures* - A group of related systems shares a common structure defined as a product-line architecture. In addition to structural properties, the product-line architecture defines the components (mandatory, optional, alternative), component interrelationships, constraints, and guidelines for use and evolution in building systems in the product-line. This architecture must support interoperability and component sharing with systems developed outside the product-line. A new system is built by using the technical reference architecture to produce a system architecture from which an implementation is constructed.

- *Development using product-line assets* - New systems are composed, adapted, or generated by populating a system architecture derived from the technical reference architecture. To the greatest degree possible, the system architecture uses existing product-line assets. This approach to development includes formal tracking of the product-line assets and identification of opportunities for reuse of the assets in other product-lines. The new system architecture and any developed or modified assets become core assets for future development in the product-line.

The product-line assets and environments are key to development of range products. They also define variations among range products. Table 1 lists some of the factors that contribute to potential variation.

Table 1. Cause of Variation Among Range Systems

Source of variation	Description
System target environment	• Customer sites • Interfacing systems • Use of specific products • System workload • Operations and logistics
Target support environment	• Development facilities • Prototyping facilities • Maintenance facilities • Integration and test facilities
Customer/user	• Organizational components • Policies, guidelines & standards • Resources • Tools and facilities • Training level & support • Technology transition support
Product-line limitations	• Operational capabilities • Performance constraints • Alternative algorithms, models or implementations • Information representation
Organizational processes	• Business/mission need analysis • Life-cycle process • Business process reengineering

	• Quality assurance

Working with the User

Range users work with the Range Product Development Groups to define operational requirements and deploy systems using product-line assets, as well as their own components. Users and test and training organizations may also rely on the product-line organization to provide domain expertise in key technology areas, such as radar, communications, and network control, rather than maintaining organic expertise in every area. Developing the expertise within the Range Development Organization and the assets that embody that expertise may be funded through pooling funds across T&E organizations or by direct core funding of the Development Organization. Under the product-line concept, the Range Product Development Groups are the designated developers of individual range products and work with the other groups within the Range Development organization to sustain the product-line and its assets. Figure 4 illustrates responsibilities of the Range Product Development Group and the range user.

Figure 4. Range User and Range Product Development Group Responsibilities

The Product-Line Approach allows early demonstration of capabilities to the user through a baseline system supporting rapid prototyping and existing products in the product-line. This early demonstration informs the user of:

- How other products look (i.e., capabilities, structure, performance characteristics, etc.),
- The bounds of tailoring,
- How requirements should be analyzed and how to manage expectations, and
- The areas of risk, i.e., those not currently covered by the product-line.

Through demonstration, the user can then determine whether the PLA will be sufficient to meet all or a subset of the user's needs.

TAILORING THE PRODUCT-LINE ARCHITECTURE

Systems developed by ranges for range users involve different groups depending on the specific system requirements for that acquisition. The Range Product Development Groups interact with other groups in their Range Development Organization. The relationship between the Product Development Groups and the other organizations involved in product-lines involves the following interactions:

- The ranges or range users provide the system requirements for the production of systems in the product-line.
- The Architecture Group works with the Product Development Group in tailoring the architecture and determining its evolutionary path.
- The Support Group works with the Product Development Group to tailor and install the execution environment.
- The Component Asset Group supports tailoring of assets and during product development, evolves existing assets or identifies and qualifies new assets.

The Range Product Development Group, working with the other product-line organizations, performs the three key tasks of product-line production:

- Support definition, use, and maintenance of product-line architecture,
- Evolve, and maintain product-line assets, and
- Produce application systems (including systems that integrate across product-lines).

The product-line architecture is driven by system requirements, thus establishing the design structure for systems in the product-line. The two-headed "Architecture" arrow in Figure 5 indicates that the specific needs of a system in the product-line will influence evolution of the architecture. The assets satisfy particular functional capabilities common to systems in the product-line or support some aspect of product development (e.g., testing, documentation). They supply the key components used in building systems in the product-line. The two-headed "Assets" arrow indicates that system needs will also influence the evolution of product-line assets. The third task is the development and production of application systems. Application systems are the successful integration of product-line architectures and assets together with any unique, newly-developed, or identified commercial components necessary to fulfill a particular need of the system. These unique components become candidates for the asset base for future product-line development.

Developing Systems with Product-line Assets

Figure 5 shows the process of delivering a software system or product. When a new product need is identified, the appropriate Range Product Development Group will work with a range or range user and, through application engineering, produce a system meeting that need. Alternatively, the range may task the Range Development Organization to provide assets to a contractor or directly to a user organization for development. The alternative means of product development offer tremendous flexibility to the user, yet retain the structure and consistency of the PLA and avoid unnecessary duplication of effort and expenditure of scarce resources.

Figure 5 illustrates that Product-lines A and B are established within the Range Product Development Group. Product-line A has two existing products, A1 and A2. The range and range user have teamed with the Product Development Group and the Architecture Group to understand the existing products within the product-line, the range of capabilities offered by Product-line A, and the ability of the Range Product Development Group to tailor product-line assets in order to determine suitability of the product-line for their new needs.

Figure 5. Product-line Systems Production Approach

The new product, labeled A3, is developed mainly by integrating reusable assets plus new software written specifically for A3, in accordance with a product-line architecture asset. The asset technology base represents the core competency, or product-line knowledge of the engineering center. The asset group assists the product-line center in identifying through domain engineering new assets not currently in the asset technology base. These assets can support the development of A3. The asset group also helps determine whether custom software written for A3 should become assets for future system production in the product-line.

Figure 6 shows the use of a common architecture to integrate alternative product-line assets in creating products for product-line A. Product A1 in the figure uses a telemetry asset and two communication interfaces, as well as other components not part of the current asset base. The new Product A3 uses a different telemetry asset and one of the same communications assets. Both share the common TENA architecture asset.

Figure 6. Building Systems from Product-line Assets

The following table summarizes the responsibilities of organizational elements.

Table 2. Responsibilities of Product-line Organizations

Element	Primary Roles and Responsibilities
Range User	• Defines and prioritizes user needs and clarifies requirements • Uses delivered systems
TERC, TIRIC, TERIB	• Establishes policy for product-line systems approach; policy for integrating across product-lines and interoperability • Ensures that all programs are identified • Approves identification of product-lines • Identifies and reserves funds for product-line creation and development • Approves each system to be developed under the Product-Line Approach
Range	• Manages system acquisition and development • Serves as the primary interface to users and between other product-line groups • Supports product-line identification • Uses product-line definition to assist in dialog with user for deriving operational

	requirements for systems • Analyzes prototypes • Validates prototype results, where appropriate

Table 3. Responsibilities of Product-line Organizations (Cont'd)

Element	Primary Roles and Responsibilities
	• Determines which, if any, of the original requirements can be tailored to conform to product-line standards • Performs acceptance testing of delivered systems • Develops plans for integration across product-lines • Manages deployment and installation
TENA Architecture Group	• Establishes, monitors, and improves the processes used in the Product-Line Approach • Identifies product-lines with ranges and range users • With Product Development Groups, ranges, and range users, defines and maintains the architectures for range systems • With range development groups, supports the evolution and reengineering of legacy systems for conformance to product-line architecture • Defines standards and methods for validating conformance with architectural definitions; responsible for "building permits" and certifying conformance
Component Asset Group	• Develops, procures, and evolves software (including COTS software) for product-lines and for product-line assets; configuration management

	• Supplies domain expertise in key product-line technology areas
Product-Line Support Group	• Qualifies environment products against product-line architectures • Identifies enterprise-wide development and execution assets (from COTS, GOTS, product development groups) • Provides a repository for test and training use
Product Development Group	• Integrates and delivers systems • Tailors the product-line architecture and components through specialization, and custom development per range and range user requirements

With the exception of the TERC, TIRIC, and TERIB, the product-line organizations shown in Table 2 are a mix of Government and contractors. These organizations exist to support the PLA. Under this structure, each product development group may be supported by contractors that produce products in specific product-lines.

PRODUCT-LINE DEVELOPMENT PROCESSES

Product-Line development is based on technical and system architectures. Products will use assets at all levels of the architecture:

- *Infrastructure assets* - These are the assets which comprise the technical reference architecture and provide services for component connection and execution.
- *Shared application assets* -Together with the infrastructure, these assets provide the common application frameworks for all range applications. They will satisfy requirements for the system architectures.
- *System architecture assets* - These assets provide the structure for specific types of ranges, e.g., OAR, ISTF. They utilize infrastructure and shared application assets and will be customized for specific facilities using range-specific assets.
- *Range specific assets* - Ranges will develop assets to support unique requirements for that range. They will be built using common infrastructure, shared application assets, and system architecture components.

The following sections provide brief examples of the processes to be used in creating large-grain component assets and using those assets to create range products. The examples are for open air ranges.

Developing Assets

Shared application assets will be developed by the Component Asset Group illustrated in Figure 2. This development will be derived from the general product-line description for open air ranges developed by the Architecture Group. The products of this asset development will include large-grain components. The process is commonly referred to as domain engineering.

For open air ranges, the Component Asset Group will follow these steps:

- From the product-line description for open air ranges, the developers will understand the key domains for range assets. These domains are actually recognized areas of expertise within the range community: telemetry, displays, situational awareness, and others. The Component Asset Group must establish the connections and relationships between these domains and also scope their bounds of applicability. During this phase, the Component Asset Group will also determine which areas are appropriate for common application support and which are range-specific.

- Within each domain, the Component Asset Group will produce a domain model to refine their understanding of the domain and then define the domain specific requirements. This understanding will define areas of commonality across ranges and those that will differ. The domain model will be represented in the form of object, feature, and behavior models. The component group may also build and test prototypes within the domain.

- Each component asset will have its own architecture derived from the technical reference architecture taking into account common and differing areas identified in the domain model. The Component Asset Group will develop the architecture, partitioning domain requirements to subsystem and object level and defining the connections. The architecture will also define external interfaces to the reference architecture, to other component assets, and to applications that will use the component.

- Component implementation will develop detailed design of the component assets. The implementation must define mechanisms for handling variation of use of the components. Variation techniques may include parameterization, inheritance, or generation. These and other implementation methods are fully explored in [Jacobson, 1997].

Figure 7 illustrates this process for creating a component asset.

Figure 7. Example Process for Asset Creation

Developing Range Products

The layered architecture provides the basis for development of range products for a specific range, as shown in Figure 8. If a range requires a combination of radar tracking, telemetry, and display, its approach might be as follows:

- Use communication services from infrastructure to provide connectivity to telemetry, radar, and displays.
- Integrate common application components through infrastructure services along with any range-unique components.
- Use application interfaces to common and range-unique components for implementation

of range product.

Figure 8 . Product Architecture for Range System

The Product-Line Approach is not a case of "one size fits all." There may be circumstances where the approach should not be followed because of costs, scheduling, performance, capability, or insufficient commonality. All product-lines may not be of the same maturity, so the procurement organization must consider risk factors in making a product-line decision. Finally, a new set of requirements may fall outside the bounds of any existing product-line. The Range Development Organization must then determine if this should be a new area for continuing work and whether establishing a new product-line is feasible. All of these factors would be considered as part of the business analysis.

Solving this problem and implementing the solution described above is more substantial than the creation of an architecture. It requires a systematic and comprehensive approach (i.e., the PLA) to marshal existing resources and identify additional methods to lower the costs of providing test and training capability. Key to this is strong management support and the identification of a champion who will assume responsibility for managing and facilitating the effort.

SELECTING A PRODUCT-LINE APPROACH CHAMPION

A Product-Line Approach will not be realized by creating a set of loosely related programs that attack selected specific (or overlapping) aspects of the problem. What is required is a systems perspective and a comprehensive plan that identifies all of the measures necessary to success and the means of insuring their accomplishment. The champion must be the owner of this plan and animate available resources in concert with each other and according to plan. *The architecture is the key element in that plan, but not the only one*.

Providing for the existence of an architecture is not enough to insure that a product-line will result. It takes a concerted, well- coordinated effort to overcome technical, cultural, political, and programmatic obstacles.

ARCHITECTURE-BASED DEVELOPMENT

Results are predicated on the use of architecture-based development. Much is implied by this approach to system design. Fundamental is the establishment of a development process centered on a system architecture that is applied to the development of the system in a prescriptive manner.

The architecture-based development approach requires the assembly of a number of related elements that are used to manage, design, implement, and test the system. These include a set of program plans (program management plan, systems engineering management plan, software development plan, configuration management plan, test and evaluation plan, integration plan, etc.), the architecture description document, a set of architectural templates or tools that automate the representation and use of architectural templates, typical development tools (CASE support for detailed design, CM tools, compilers, GUI builders, etc.), and documentation tools. The program plans need to recognize that this approach requires unique support to make efficient use of resources. Plans should establish the management infrastructure and reporting elements similar to the structure of the architecture. The processes of estimation and tracking should be directly keyed to this structure. The program schedule needs to reflect a commitment of resources and time during early phases of the program to development and validation of the architecture. This usually includes sufficient prototyping to effect validation of architectural decisions and discover detail about unprecedented parts of the system.

A number of products are assembled from this process. One group is a set of composition tools procured and/or created to generate system instances from components that are compliant with the architecture. Components are assembled from repositories of parts created by developers. They include infrastructure components that provide system-wide services and structural backbone used by application components. They handle such issues as scheduling, message management, time management, security, marshaling operating platform services, synchronization, etc. Repositories also include application components that provide specific functionality related to the domain and system functional requirements.

As part of the architecture development a number of artifacts are created that are generally considered part of the architecture. These include the basic conceptual structure itself which captures and abstracts the fundamental repeating patterns of the solution domain, templates or tool definitions used to create instances of the structural types provided by the architecture, standards applied to subsequent, lower-level design and implementation, and guidelines for the use and adaptation of the architecture. The tangible parts of the architecture are the collection of infrastructure components and high-level class definitions for application components.

Architecture-based development represents a paradigm shift for most system developers. The history of real-time systems is provision of a high-level view of the system oriented primarily to the system's hardware with software modules situated within hardware assets. This is commonly the extent of the architecture. Diagrams of software component breakdowns, when provided, are usually somewhat arbitrary and fail to capture the underlying natural structure of the problem space. Developers are free to implement within broad guidelines, and the true system architecture emerged as a derivative property of the system from that chaotic process. When documented, the architecture description is often the as-built condition but of questionable accuracy and completeness There is a considerable effort of training required of developers and managers to transition this approach and institutionalize its employment. The transition needs to be planned, a concept of operations must be created, user guides and tutorial examples developed, and acquisition practices must be adapted to recognize the need to reward contractors for performing the right kinds of activity early in the program.

IMPACT OF TRANSITION TO A PLA STRATEGY

The transition to a product-line strategy requires significant change in existing organizations. Any plan for transition must address the impact of change on organization, management, and acquisition elements.

Organization

The Product-Line Approach will require special attention to bring together core competencies from across existing organizational structures. There appears to be significant redundancy of personnel and skills within the current test and training organizations. Product-line organizational restructuring will enable concentration and sharing of personnel and skills.

While the establishment of a product-line philosophy will affect product users, Program Executive Offices (PEOs), contractors, and support organizations, the principal impact will be on the direct users of the product-lines: the ranges and range users. The Range Development Organization will coordinate the interaction of users, the Architecture

Group, and the Product Development Groups for proposed systems. Ranges will rely on the Range Development Organization for technology expertise and development and on the Product Development Groups, together with the Architecture and Support Group, to establish the specifics for system implementation and for configuration management as they affect the product-line. During product sustainment, the Range Development Organization will review the existing product-lines and architectures and establish a reasonable maintenance/ upgrade/enhancement plan for the product.

Management

New incentives will be needed to support the management and use of a PLA. Organization elements of key importance to ranges will be smaller than they are today, but no less critical. The following steps will help manage the technological changes that come with adopting a product-line approach:

- Promotion and reward potential must be addressed in the new structure.
- General cultural changes will be needed at all levels. Management must drive these changes, even when they are the most affected.
- Organizations will need to learn to get their job done, i.e., field a system, by relying on support and assets from other parts of the organization. Not all aspects of a program will be under the control of one manager.
- Product-line orientation requires sharing of responsibilities and resources and is impossible in a stovepiped organizational structure.
- A managed process for product-line development will support certification of system conformance to the product-line architecture and successful use of product-line assets.

Acquisition

Systems need to be acquired through methods that encourage the use of existing product-line infrastructure and directly support the maintenance and upgrade of the infrastructure to support future needs. The current acquisition process funds software-intensive efforts on a program-by-program basis, with minimal funding allocated to product-line infrastructure. More investment is needed in support of a series of systems based on a common infrastructure. The test and training community can address many near-term changes with local acquisition strategies. These local acquisition strategies include:

- Coordination of development activities among ranges,
- Elimination of redundant development, and
- Use of funds to further the development of the product-line for the benefit of all the contributing programs.

Test and training can also pool funds from all the ranges that fall within a product-line to pursue product-line development. One program may be established to manage the common infrastructure. Other program offices would contribute software assets to evolve the product-line. Product-line approaches also support procurement reform initiatives by taking advantage of commercial practices, existing COTS software products, and standardization of newly developed product-line components that can be reused across systems.

One of the initiatives being fostered by the Director, Defense Test, Systems Engineering, and Evaluation (DTSE&E) to increase T&E's value to the acquisition process is the Simulation, Test and Evaluation Process (STEP). [Sanders, 1997] [TENA, 2]. STEP has expected payoffs in program cost savings, shorter development schedules, increased productivity, and improved mission performance. The Product-Line Approach is a methodology for supporting STEP across the test and training domains and ensuring achievement of major reform goals and objectives.

Management must ensure that every new program is examined for similarities with existing systems in mission and underlying functions. The goal is to focus new development on unprecedented areas and reuse product-line assets as much as possible. Reuse of assets includes much more than software components. Design, architecture, requirements, and models are all assets for reuse. Acquisition strategies will need to ensure that every procurement leverages past investments to the fullest and contributes to assets used in future efforts.

Because of the increased focus on assets (including non-code assets, such as architectures) and their management for use across more than one system, product-lines will bring ownership and liability issues to the forefront. The current acquisition regulations define a range of options for software and data rights, ownership, and liability issues that are most likely sufficient to address product-line implementation. While the current acquisition guidelines provide a sound framework for dealing with issues of ownership and asset management, additional guidance will be needed on their application to product-line concepts.

Ownership of assets within the PLA is a key question. A Government organization should own the product-line architectures or at least the right to use the architectures. The product-line and the non-COTS software built to field a system may be government- or industry-owned. One model for the organizational structure of product-line assets is government and industry:

- *Government* - Defines and owns Government Purpose License Rights (GPLRs) to product-line architectures to define component structure, connections, and constraints for a class of systems.

- *Industry* - Develops components driven by market need and integrates components as part of a system within a product-line. The government obtains GPLR while the contractor retains commercial rights.

SUPPORT STRATEGY

A basic element of the product-line strategy is the continued maintenance and enhancement of the product-lines and the corresponding architectures. The entire Range Development Organization will cooperate in this effort, with the Architecture

Group taking the lead. Architecture maintenance and enhancement is the primary responsibility of the Architecture Group, which will lead architectural assessments to determine the needs for enhancement or, possibly, a new architecture. The Component Asset Group is responsible for actual enhancements to product-line components and ensuring that new versions of COTS products are integrated into the product-lines. The Support Group is responsible for maintaining the Product List supporting the product-lines and for working with vendors to coordinate maintenance of their products. Updated products are provided to the various customers/users according to maintenance/upgrade agreements established at the initiation of a system acquisition. The maintenance and support of the product-line architectures and components is a natural consequence of the product-line development strategy.

By using the product-line systems approach, organizations will deploy systems faster, at a lower cost, and with fewer Government and industry resources. Systems will be even more reliable because they will use common components with high reliability and proven performance. Training will be improved since common components will reduce the amount of training currently needed when transitioning between command and control systems. More commercial components will be available because industry will identify a larger market for their products when used across similar systems. Upgrades of components will also be promoted as industry recognizes a new market for their enhanced products.

The successful implementation of a product-line systems approach presents challenges and barriers that are significant but surmountable. These include:

- *Cultural* - Product-line strategies mean organizations and managers have less direct control over their product developments and increased dependency on other organizations to understand their requirements and provide acceptable solutions. Giving up this control and the necessary dollars to support product-line technology and application development may be difficult.

- *Strategic planning* - Product-line planning is not only a management process that links related systems. The Range Development Organization must consider the long-term needs of users and the ability to build products for those users. They must take an enterprise-wide look at existing and planned products and look several years into the future in planning for product-lines. The future year development plans should focus attention on product-lines as the means to satisfy the plan.

- *Need for tradeoffs* - The Product-Line Approach presents a tradeoff for the user between "build me the exact system I want" and "build me a system almost like what I want using the product-line, saving on costs and time."

- *Resource ownership* - Who will "own" the product-line components? How will they be funded? These issues require transitioning from program-focused acquisition organizations and budgets to more commercial-like product organizations and budgets.

- *Recognition and reward* -The current acquisition system focuses on recognition and rewards for personnel on delivered systems. Use of product-line strategies also necessitates a shift to rewarding and advancing personnel for broadening the utility of products and facilitating their use within and across product-lines.

- *User interface* - Users will experience close ties to the development organization within the Product Development Groups. They should experience greater responsiveness through improved needs definition, refinement, and early demonstration. However, operational users must adjust to having more than the program managers as their dependency links to successful system upgrades or developments. This should not be difficult since users today are regularly dependent on a variety of sources for successful systems deliveries.

- *Effects of technological change* - The transition to a Product-Line Approach will mean significant changes in our current way of doing business. We must plan for the effects this will have on the individuals who must carry out the transition and also on those who will be operating under the new approach.

It is strongly recommended that the CTEIP program plan an incremental, evolutionary development centered on creation of the Product Line Approach. It is the basis for early implementation and prototyping, estimation of performance and quality characteristics, and the evolution of the system from a small functioning core to the full system. There are many advantages to the use of this approach, including the ease of working from a small core and adding functionality progressively, early validation of system wide design decisions, promotion of parallel development, easier integration with much greater robustness of the system concepts, and facilitation of estimation of system performance, quality, and cost.

After the initial approach is established, the test and training community should expedite the following actions:

- Create the Architecture Group, Component Asset Group, and Support Groups. These organizations are not created for every new product-line, only when existing groups cannot or should not support a new product-line. The Product Development Groups will also be created to support product-line production.

- Analyze the current product mix to identify potential product-lines. The analysis should review the current status of programs and the plans for future evolution. The organization must consider its current and anticipated customer base. It is possible that ongoing programs will need resources and/or relief in program milestones to assist in the development of the asset base and transition to product-lines.

- Define the assets for product-line development according to desired product variety and customer needs. The Range Development Organization must identify the processes that are part of asset creation including domain engineering, architecture description and assessment, and reengineering to deal with legacy systems. The Architectures Group will be responsible for defining and monitoring these processes for the Range Development Organization.

Along the transition path, the test and training communities should look for new or ongoing programs that can immediately contribute to the Product-Line Approach. The Architecture Group should seek ways to develop architectures and work with ongoing programs to make sure these systems produce common product-line assets, or that the components they produce can become product-line assets.

The Product-Line Approach will not fit all system solutions, but should make a significant improvement in the timeliness, cost, and reliability of systems found on MRTFB and Training ranges and facilities, such as HITLs, ISTFs, M&S programs and others suitable for its application. Business analysis must determine where and how to apply the Product-Line Approach.

Use of product-line strategies will enhance reliability of performance, schedule, and cost estimates. This dependability in delivery of systems will be achieved through elimination or narrowing of the "bounds of uncertainty" that accompany any estimating process. Where proven components are incorporated into the design and estimating process, we immediately improve our confidence levels surrounding performance, schedule, and cost expectations. Our assessments of risks and the uncertainties associated with resolving those risks are more narrowly bounded because of the improved ability to control the expectations associated with use of product-line components. Metrics are more readily available to assist in the estimating process and establishment of schedule and performance parameters.

Adoption of the Product-Line Approach requires thorough business analysis and careful planning. The test and training communities must assess the business opportunities to ensure the appropriateness of adopting a Product-Line Approach. During transition, the organization must carefully monitor progress and make sure that product-line groups are giving effective support.

Contractors work with several related programs to develop a common architecture and other assets. While it is not necessary for the Government to own all the assets in the product-line asset base, it is necessary to have appropriate access to them. The acquisition and ownership policy for product-line architectures is under investigation by several groups within the DoD.

Product-line development evolves naturally from applying fundamental engineering concepts to meeting recurring needs. Recurring requirements provide the potential for economies of scale and reuse. Doing the job better, faster, and cheaper requires a focus on efforts that reduce the variable costs associated with system development and the total life cycle.

BoOD	Board of Operating Directors
CASE	Computer-aided software engineering
CM	Configuration management

COTS	Commercial-off-the-shelf
CONOPS	Concept of Operations
CTEIP	Central Test and Evaluation Investment Program
DISA	Defense Information Systems Agency
DMSO	Defense Modeling and Simulation Office
DoD	Department of Defense
DTSE&E	Director, Test, Systems Engineering & Evaluation
GOTS	Government off-the-shelf
GPLR	Government Purpose License Rights
GUI	Graphic user interface
HITL	Hardware-in-the-loop
HLA	High Level Architecture
IL	Integration laboratory
ISTF	Installed system test facility
JPO	Joint Project Office
JSIMS	Joint SIMulation System
M&S	Modeling & Simulation
MF	Measurement facility
NUWC	Naval Undersea Warfare Center
OARO&M	Open Air Range Operation Maintenance
PEOs	Program Executive Offices
PLA	Product Line Approach
STEP	Simulation, Test and Evaluation Process
T&E	Test and Evaluation
TENA	Test and Training ENabling Architecture
TERC	Test and Evaluation Resource Council
TERIB	Test and Evaluation Reliance Investment Board

TIRIC	Test Instrumentation Resource Investment Council
TRA	Technical Reference Architecture

[Brownsword, 1996] Brownsword, Lisa and Clements, Paul, "A Case Study in Successful Product Line Development," Software Engineering Institute, Carnegie Mellon University, CMU/SEI-96-TR-016, October 1996.

[Dikel, 1997] Dikel, David, et al., "Applying Software Product-Line Architecture," *IEEE Computer*, August 1997, pp 49-55.

[Jacobson, 1997] Jacobson, Ivar; Griss, Martin; Jonsson, Patrik., *Software Reuse : Architecture Process and Organization for Business Success*. New York, ACM Press ; Reading, MA, Addison Wesley Longman, 1997.

[Lightning Bolt] Lightning Bolt Initiatives, A Series of Eleven Air Force Acquisition Reform Initiatives. See website www.safaq.hq.af.mil/acq_ref/#lightning bolt.

[Macala, 1996] Macala, Randall R., et al., "Managing Domain-Specific, Product-Line Development," *IEEE Software*, vol 13, no. 3, May 1996.

[Northrop, 1997] Northrop, L.M., "Product Line Practice: The Goal and The Current State," Software Engineering Institute, Carnegie Mellon University, August 1997.

[Perry, 1994] Perry, William J., Secretary of Defense. "Acquisition Reform: A Mandate for Change," Washington, DC: Department of Defense, 1994.

[Sanders, 1997]. Sanders, Patricia, "Test and Evaluation to Meet the Modernization Challenge," Director, Defense Test, Systems Engineering, and Evaluation, *ITEA Journal*, pp. 29-32, March/April 1997.

[STEP, 1996] Simulation, Test, and Evaluation Process (STEP) Guidelines, Draft document from Director, Test, Systems Engineering and Evaluation, Department of Defense, December 1996 (UNCLASSIFIED).

[TENA,1] Baseline Project Report, Volume IV, Technical Reference Architecture, 1997.

[TENA,2] Baseline Project Report, Volume X, Simulation, Test, and Evaluation Process

CELSIUSTECH - SHIPBOARD COMMAND AND CONTROL

CelsiusTech Systems AB is a Swedish naval defense contractor and one of Sweden’s leading suppliers of command and control systems. Current customers include the navies of Sweden, Denmark, Finland, Australia, New Zealand, and Pakistan.

Until the mid-80's, most of CelsiusTech's work was project-oriented, with individual customers, and each system viewed as a new start. In December, 1985, this approach changed when the company signed contracts for simultaneous delivery of two separate systems. These systems differed from previous systems in several ways; they were

- Larger in size,
- Greater in complexity of requirements, and
- Fixed-price contracts.

CelsiusTech's previous C3I project experience indicated the potential for significant problems, including

- lack of predictable and timely integration,
- continual requirements changes,
- cost and schedule overruns, and
- shortage of software engineers in Sweden.

These new challenges forced CelsiusTech to define a new business strategy for satisfying the two contracts. Their decision was to create a common architecture from which to design and implement both systems. This architecture would be developed to satisfy requirements known to exist from system to system within the shipboard command and control area. In short, CelsiusTech devised a product-line approach for delivery of these systems. In addition to the technical needs, from the business perspective, this approach had resulted in a product-line adaptable over a wide range of systems where systems could be implemented at reasonable cost.

CelsiusTech was willing to invest in supporting a technical strategy. This strategy would guarantee that configuration of each new system would come from product-line assets and ensure that new projects would enrich the product-line. CelsiusTech also evaluated current technology infrastructure to create a new generation of system: hardware, software, development approach, and development/execution environments. Within this approach, reuse was an enabling strategic business strategy.

CelsiusTech's first task was to develop an architecture for supporting the two concurrent contracts. This architecture was a basic layering structure, with specific layers:

- a virtual machine to hide hardware-specific interfaces,
- application support for maps, picture compilation, and ship's information, and
- mission-specific applications for target tracking, fire control, anti-submarine warfare, and electronic countermeasures.

Within each layer are collections of components that can be integrated in different combinations to satisfy individual system requirements. The architecture is now called Ship System 2000 and has been used on more than 10 different ships. Figure C-1 illustrates the layering concept.

Figure C-1 . Product-Line Architecture for Ship System 2000

The use of Ship System 2000 and the product-line concept for system development has had several notable benefits:

- Hardware-to-software cost ratio has changed from 35:65 to 80:20.
- Integration test of 1-1.5 million SLOC is handled by two staff people (compared to three-five people for 200-500K SLOC).
- Performance/distribution behavior for new systems is known before project start.
- Rehosting the product-line to new platform/OS takes three months.
- Very predictable cost and schedule are based on known requirements for product-line configurations.
- There is high customer satisfaction.

Since the successful introduction of the shipboard product-line, CelsiusTech has initiated a second product-line in ground-based air defense systems. Between these two product-lines there is significant commonality. The key factor in the successful introduction of this second product-line is the common Product-Line Approach.

HEWLETT PACKARD - MICROWAVE INSTRUMENT FIRMWARE

Hewlett Packard developed a pilot-driven policy for adopting reuse across a product-line

[Jacobson, 1997]. The need for a new approach to product delivery emerged in response to time to market pressure. Rapid production of software was the key to meeting market pressure. In addition, the company saw an increasing demand for a greater variety of products that allowed customization from standard features. The goal of customization put extra pressure on the need for an architecture to supply the structure for accommodating both the feature customization and reuse.

The initial step toward a firmware product-line was creation of an instrument product council. The council approved a pilot architecture effort that produced a common structure for firmware products and large-grain components to support the products. These components initially accounted for 30% of a typical 200K firmware system.

Hewlett Packard cites a corporate-wide tour of instrument divisions as key to their success. The tour allowed product-line architects to understand divisional needs and allowed divisions the ability to buy in to the concept. Since component developers came from the divisions, the tour gave them an understanding of what was required to meet architecture needs.

In addition to introducing the architecture, the architecture organization produced an instrument specification language and a system generator. Each of five divisions contributed one major component which the others depended upon for their system. Within two years, the effort resulted in delivery of two products. Over the next two years, 15 more products were delivered using the initial components plus those added later. The benefits are obvious:

- Reduction in time to market from 18 months to 5 months, and
- Up to 50% of the products based on existing components.

The architecture and constituent components are part of standard platforms for firmware. The microwave divisions have created a technology center to support these assets for future development.